CLAIMS

1. A method in a computer system for dispatching requests to perform services to sub-applications that use different logic models the method comprising:

providing a context for the sub-applications

receiving a request to perform a service; and

for a plurality of sub-applications,

determining whether the received request should be dispatched to the sub-application; and

when it is determined that the request should be dispatched to the sub-application, invoking a service routine of the sub-application passing the request

whereby the sub-applications share the provided context.

2. The method of claim 1 wherein the sub-applications are ordered and the invoking of the service routines is performed in the order of the sub-applications.

3. The method of claim 1 wherein the determining includes determining whether a match criteria for the sub-application matches the received request.

4. The method of claim 3 wherein the requests are HTTP requests with a URL and the match criteria is a regular expression relating to the URL.

5. The method of claim 1 including suppressing the invoking of additional service routines when an invoked service routine returns an indication to suppress the invoking of additional service routine.

6. The method of claim 1 including suppressing the invoking of additional service routines when an invoked service routine responds to the received request.

7. The method of claim 1 wherein an invoked service routine performs user authentication and indicates to suppress invoking of additional service routines when a user cannot be authenticated.

8. The method of claim 1 wherein an invoked service routine logs the received request.

9. The method of claim 1 wherein an invoked service routine logs a response of another invoked service routine.

10. The method of claim 1 wherein an invoked service routine transforms the received request from one protocol to another protocol.

11. The method of claim 1 including:

for each of a plurality of sub-applications,

retrieving initialization parameters for the sub-application;

retrieving an indication of a class for the sub-application; and

instantiating an instance of the class with the retrieved initialization parameters.

12. The method of claim 1 wherein the determining includes determining whether a match criteria in a configuration file for the sub-application matches the received request.

13. The method of claim 1 wherein the determining is performed by a match routine of the sub-application.

14. The method of claim 1 wherein a sub-application uses interaction-based model.

15. The method of claim 1 wherein a sub-application uses an action-view model.

16. The method of claim 1 wherein a sub-application uses a workflow-based model.

17. The method of claim 1 wherein the sub-applications form an overall application and wherein the provided context is an application-level context.

18. The method of claim 1 wherein the sub-applications form an overall application that is web-based.

19. The method of claim 1 wherein the request is received from a web-server environment.

20. A computer system for dispatching HTTP requests to sub-applications, comprising:

a configuration file having a class, initialization parameters, and a match criteria associated with the sub-applications;

an initialization component that instantiates an object of the class for each sub-application in the configuration file, the instantiated object being initialized with the initialization parameters for the sub-application and being provided with a context object, the context object being shared by the instantiated objects so that the sub-applications share a common context; and

a dispatcher that receives HTTP requests and, when the received HTTP request matches a match criteria of a sub-application, invokes a service routine of the instantiated object of the class associated with the sub-application.

21. The computer system of claim 20 wherein the match criteria is a regular expression relating to a URL of the HTTP request.

22. The computer system of claim 20 wherein the configuration file specifies an ordering of the sub-applications and the dispatcher invokes the service routines in the specified order.

23. The computer system of claim 20 wherein the dispatcher does not invoke any additional service routines when an invoked service routine returns an indication to not invoke any additional service routines.

24. The computer system of claim 20 wherein the dispatcher does not invoke any additional service routines when an invoked service routine responds to the received request.

25. The computer system of claim 20 wherein a sub-application is based on an interaction model.

26. The computer system of claim 20 wherein a sub-application is based on an action-view model.

27. The computer system of claim 20 wherein each of the sub-applications implement the same interface.

28. A computer system for processing request messages, comprising:

a plurality of sub-applications forming an application, a sub-application having a match criteria indicating when the sub-application should process a request and having a service routine to invoke when the match criteria indicates that the sub-application should process the request, the sub-applications using disparate logic models;

a context for the application that is shared by the sub-applications; and

a dispatcher that receives requests, evaluates the match criteria to identify which sub-applications should process the requests, and invokes the service routines of the identified sub-applications wherein an invoked sub-applications use the context.

29. The computer system of claim 28 including an initialization component that instantiates an object of a specified class for each sub-application.

30. The computer system of claim 29 wherein the initialization component accesses configuration information that specifies the class of each sub-application and any initialization parameters for the sub-applications.

31. The computer system of claim 29 including a context object representing the context and wherein the initialization component provides the context object to each sub-application.

32. The computer system of claim 28 wherein each service routine is passed a request parameter and returns a response parameter.

33. The computer system of claim 28 wherein the sub-applications are ordered and the dispatcher invokes the service routines based on the order of the sub-applications.

34. The computer system of claim 33 wherein an invoked service routine indicates that additional service routines should not be invoked to process the received request.

35. The computer system of claim 33 wherein the dispatcher does not invoke additional service routines when an invoked service routine responds to a received request.

36. The computer system of claim 28 wherein the request is an HTTP request message.

37. A computer system for processing request messages, comprising:

a plurality of service means for servicing requests, the service means forming an application, each service means having a match criteria indicating when the service means should be invoked, the service means implementing different logic models; and

dispatch means for receiving requests, evaluating match criteria to identify which service means should be invoked to process the requests, and invoking the identified service means whereby the service means share a context that is common to the service means of the application.

38. The computer system of claim 37 including an initialization means for instantiating an object of a specified class for each service routine.

39. The computer system of claim 38 wherein the initialization means accesses configuration information that specifies the class of each service means and any initialization parameters for the service means.

40. The computer system of claim 37 wherein each service means is passed a request parameter and returns a response parameter.

41. The computer system of claim 37 wherein the service means are ordered and the dispatch means invokes the service means based on their order.

42. The computer system of claim 41 wherein an invoked service means indicates that additional service means should not be invoked to process the received request.

43. The computer system of claim 41 wherein the dispatch means does not invoke additional service means when an invoked service means responds to a received request.

44. A computer-readable medium for controlling a computer system to dispatch requests to perform services to service routines, by a method comprising:

receiving a request to perform a service; and

for a plurality of service routines,

retrieving a match criteria for the service routine;

determining whether the received request matches the retrieved match criteria;

when it is determined that the request matches the retrieved match criteria, invoking the service routine for processing of the received request

whereby the service routines form an application and share a common context.

45. The computer-readable medium of claim 44 wherein the service routines are ordered and the invoking is performed in the order of the service routines.

46. The computer-readable medium of claim 44 wherein the requests are HTTP requests with a URL and the match criteria is a regular expression relating to the URL.

47. The computer-readable medium of claim 44 including suppressing the invoking of additional service routines when an invoked service routine returns an indication to suppress the invoking of additional service routine.

48. The computer-readable medium of claim 44 including suppressing the invoking of additional service routines when an invoked service routine responds to the received request.